

1. Can we disable the default web server in the Spring Boot application?

Yes, we can disable the default web server in the Spring Boot application. To do this, we need to set the `server.port` property to “-1” in the application’s `application.properties` file.

2. Spring Boot 3 Features?

Spring and Spring Boot are two of the most popular Java frameworks used by developers worldwide. The Spring team is continuously working on improving and enhancing the frameworks with each new major release. Spring 6 and Spring Boot 3 are expected to bring in significant new features and changes that will further boost development with these technologies. In this article, we embark on a journey to uncover the exciting innovations that Spring 6 and Spring Boot 3 bring to the table.

What’s New in Spring 6:

- **Java 17+ baseline:**
 - Support for Java 17 language features like sealed classes switch expressions, etc. Spring code and APIs will be updated to leverage new Java 17 capabilities where applicable.
 - Support for additional reactive stream operators and features introduced in JDK 17 like reactive stream publisher combinators. Spring WebFlux and reactive components may utilize these.
 - Spring 6 is runtime compatible with Jakarta EE 9 as well as Jakarta EE 10 APIs. This means that you can use Spring 6 to develop applications that can run on Jakarta EE 9 or Jakarta EE 10 compliant runtime environments.
 - Spring 6 is compatible with the latest web servers, such as Tomcat 10.1, Jetty 11, and Undertow 2.3. This means that you can use Spring 6 to develop applications that can run on these web servers.
- **Core Container:**
 - Spring 6 introduces support for Ahead-Of-Time (AOT) compilation. This can significantly improve the performance of your applications by compiling them into native code.
 - `PathMatchingResourcePatternResolver` uses NIO and module path APIs for scanning, enabling support for classpath scanning within a GraalVM native image and within the Java module path, respectively.
 - `DefaultFormattingConversionService` supports ISO-based default `java.time` type parsing.
- **Data Access and Transactions:**
 - Spring 6 provides JPA support for Hibernate ORM 6.1. This includes support for new features in Hibernate ORM 6.1, such as the ability to use multiple database dialects in the same application.
 - Spring 6 upgrades to R2DBC 1.0. This includes support for new features in R2DBC 1.0, such as the ability to define transaction boundaries for R2DBC connections.
 - Spring 6 removes support for JCA CCI. This is because JCA CCI is a legacy technology that is no longer actively maintained.
 - Spring 6 aligns the data access exception translation between JDBC, R2DBC, JPA and Hibernate. This means that you will see the same exception types regardless of the data access technology you are using.

- **Spring Messaging:**
 - Spring 6 Messaging supports RFC 7807 problem details, which is a standard for returning error responses that include information about the error, such as the error code, message, and more.
 - Spring 6 Messaging also includes new features for Spring Integration, such as support for the Reactor Netty library and the ability to send and receive messages asynchronously.
- **Spring MVC:**
 - Spring MVC 6.0 now has built-in method validation support for controller method parameters with `@Constraint` annotations. This means you no longer need `@Validated` at the controller class level to enable method validation via AOP proxy.
 - Spring MVC 6.0 has improved support for reactive web applications. This includes support for the Reactor Netty library and the ability to handle requests asynchronously.
- **Spring WebFlux:**
 - Spring WebFlux will upgrade its Reactive Streams implementation to version 2.1 for improved interoperability.
 - Spring WebFlux is expected to add initial support for the new HTTP/3 protocol for reactive and low-latency communication.
- **Testing:**
 - Spring 6 Testing has improved support for reactive testing. This includes support for the Reactor Netty library and the ability to write tests that are asynchronous and non-blocking.
 - Spring 6 Testing also includes new features for Spring Boot Test, such as support for the Reactor Netty library and the ability to write tests that are asynchronous and non-blocking.

What's New in Spring Boot 3:

- **Java 17 Baseline and Java 19 Support:**
 - One of the most significant changes in Spring Boot 3.0 is the requirement for Java 17 as the baseline version. This means that Spring Boot 3.0 will only run on Java 17 or higher. Java 17 is the latest long-term support (LTS) version of Java, and it includes a number of new features and improvements.
 - Spring Boot 3.0 also supports Java 19, which is the latest release version of Java. However, Java 19 is not an LTS version, so it is not recommended for production use.
- **Dependency Management:**
 - Spring Boot 3.0 uses the Jakarta EE 9 specification as its foundation. This means that all of the dependencies that are managed by Spring Boot have been updated to use the Jakarta EE 9 packages.
 - If you are upgrading an existing Spring Boot application to Spring Boot 3.0, you will need to update the dependencies in your `pom.xml` or `build.gradle` file to use the Jakarta EE 9 packages.
- **Native Images with GraalVM:**
 - GraalVM Native Image is a technology that can be used to compile Java applications ahead of time (AOT) into a native executable. This can significantly improve the performance of Java applications.
- **Improved `@ConstructorBinding` detection:**
 - Spring Boot 3.0 can now automatically detect `@ConstructorBinding` annotations on beans. This can make it easier to create beans that are injected with dependencies.

- **Log4j2 Enhancements:**
 - Spring Boot 3.0 includes a number of enhancements to Log4j2, the logging framework used by Spring Boot. These enhancements include support for asynchronous logging, improved performance, and better security.
- **Micrometer Updates:**
 - Spring Boot 3.0 includes updates to Micrometer, the metrics library used by Spring Boot. Spring Boot Actuator metrics collection and reporting will be improved. Metrics like memory usage, garbage collection, etc will be collected out of the box.